

## POSZTPROCESSZOR FEJLESZTÉSE KIBER-FIZIKAI RENDSZERBEN

Szunyi Attila<sup>a</sup>, Andó Mátyás<sup>b\*</sup>, Safranyik Ferenc<sup>b</sup>

<sup>a</sup> ELTE, Informatikai Kar, Savaria Műszaki Intézet, Duális gépészmérnöki BSc, 4. évf.

<sup>b</sup> ELTE, Informatikai Kar, Savaria Műszaki Intézet, egyetemi docens

### ABSZTRAKT

CNC esztergagép posztprocesszorának létrehozása lényegesen hatékonyabb kiber-fizikai rendszerben (CPS). Egy Akira-Seiki SL25MC típusú CNC esztergagép posztprocesszorát Sinutrain szimulációs szoftver segítségével hoztuk létre Edgcam-ben. A kidolgozott eljárással csökken a szintaktikai hibák kiszűrésére fordított idő. Jelentősen nő a CNC gép rendelkezésre állása, mert a tesztelési feladat túlnyomó része nem a valós gépen történt. Ráadásul az esetleges hibás programkódok nem okoznak ütközést a CNC gépen. Megállapítottuk, hogy a valós gépen történő tesztelés azonban nem hagyható el, itt kiemelt figyelmet kell fordítani a mozgási sebességekre és forgásirányokra.

**Kulcsszavak:** *posztprocesszor, CAM, CNC, CPS, kiber-fizikai rendszer*

### 1. Bevezetés

Az informatika fejlődésével és térnyerésével egyre jobban megváltozott a gépészet. A kézi rajzok helyét teljesen átvették a 3D-s CAD modellek és az azokból képzett rajzok. A megrajzolt alkatrészek gyártási módszereire is hatással van az informatikai fejlődés, azonban még nem ennyire látványosan. A forgácsolás terén leginkább a szerszámgépek vezérlésére volt nagy hatással. A CNC – számítógép által vezérelt – szerszámgépek már fél évszázada léteznek, szinte az összes iparágban elterjedtek. Ezek programozására számtalan módszer létezik.

A legegyszerűbb, amikor lapon írják a programkódot, majd beírják a szerszámgép vezérlőjébe. Ez az egyszerűnek tűnő, olcsó módszer azonban számos hibalehetőséget rejt magában. A korszerű szerszámgépeken, közvetlenül a vezérlőben létrehozni a programot már egy fokkal előnyösebb és gyorsabb, azonban sokszor a gyártási körülmények nem engedik meg, illetve adott esetben a programozás idejére a gép kiesik a termelésből is. Ennek kiküszöbölésére az egyik megoldás a CAM – Computer Aided Manufacturing, azaz számítógéppel támogatott gyártás – szoftverek használata. A CAD rendszerek széleskörű alkalmazásának köszönhetően napjainkban minden gyártandó alkatrész 3D modellje rendelkezésre áll. Ezeket a modelleket a CAM szoftverekbe importálva létrehozhatók rá a megmunkálási programok egy átlátható, ergonomikus környezetben, akár a szerszámgéptől távoli irodában is.

Az alkatrész összes fizikai mérete és geometriai jellemzője, amelyek a programhoz szükségesek, ilyenkor a modelltől származnak, így azokat a program írásakor nem szükséges megadni. Ezáltal csökken a hibalehetőségek száma a programozás során. A CAM szoftverek használatának számos előnye van még, azonban nem szabad megfeledkezni a hátrányairól sem. Napjainkban sokféle CAM szoftver és még több fajta szerszámgép típus van. A gépgyártók pedig a termékeiket különféle

vezérlőkkel kínálják. Ahhoz, hogy a CAM szoftverben képesek legyünk programkódot generálni, nélkülözhetetlen egy posztprocesszor. Ez tulajdonképpen egy számítógépes szoftver, ami megteremti a kapcsolatot a CAM szoftver és egy konkrét szerszámgép között.

A programkódok létrehozásánál tehát mindig figyelni kell arra, hogy milyen típusú vezérlővel rendelkezik az adott szerszámgép. A vezérlők többségéhez rendelkezésre állnak számítógépen futtatható vezérlőszimulátorok, amelyek segítségével irodai környezetben szimulálható a vezérlők működése.

A CNC szerszámgépek programnyelve az ún. „G-kód”. Ezeket az ISO 6983-as szabvány tartalmazza. A szerszámgépek ezen a nyelven kapnak minden lényeges utasítást: hova és hogyan mozogjanak, milyen gyorsan, milyen fordulatszámon... Emellett minden CNC gép rendelkezik valamilyen vezérlővel. A több évtizedes fejlesztés során számos vezérlő gyártó és típus alakult ki. Ezekhez kialakultak vezérlőspecifikus G-kódok is, amelyek megnehezíthetik a gépek programozását, ugyanis egyes programkódok teljesen más funkciót láthatnak el egy Siemens vezérlőn, mint például egy Fanuc-on [1, 2].

Egy posztprocesszor mindig egy konkrét szerszámgép és egy konkrét CAM szoftver közötti kapcsolatot teremti meg, tehát mindig egy egyedi fájl. Az általunk készített posztprocesszor az egyetem gyártástechnológia laborjában található Akira-Seiki esztergára készült, Edgcam-ben. A cikk a posztprocesszor létrehozásával, fejlesztésével és tesztelésével foglalkozik, modern informatikai környezetben. Érintjük a posztprocesszorhoz illeszkedő szerszámgép CAD modelljének szerkesztését és a megfelelő vezérlő szimulációs környezetének használatát is. A létrehozott posztprocesszor működését teszteltük valós környezetben is. A cikkben a fejlesztési lépéseket időrendben dolgozzuk fel.

## 2. Gép és módszer

A fejlesztési folyamatot a Akira-Seiki CNC esztergagép kapcsán végeztük el. Ez az SL25MC típusú gép egy C-tengelyes eszterga, melynek vezérléséről egy Siemens Sinumerik 828D típusú vezérlő gondoskodik. A szerszámgép a vezérlőjével az 1. ábrán látható.

A fejlesztési folyamatot a következő szoftverek támogatták:

- CAD – Autodesk Inventor Professional 2018,
- CAM – Edgcam 2018 R2,
- Szimulációs szoftver – Siemens Sinutrain 4.5.

A posztprocesszor fejlesztés leghosszabb és legdrágább folyamata a tesztelés. Ennek kiküszöbölésére a tesztelés túlnyomó részét kiber-fizikai rendszerben végeztük. Az új fejlesztési folyamat felépítését



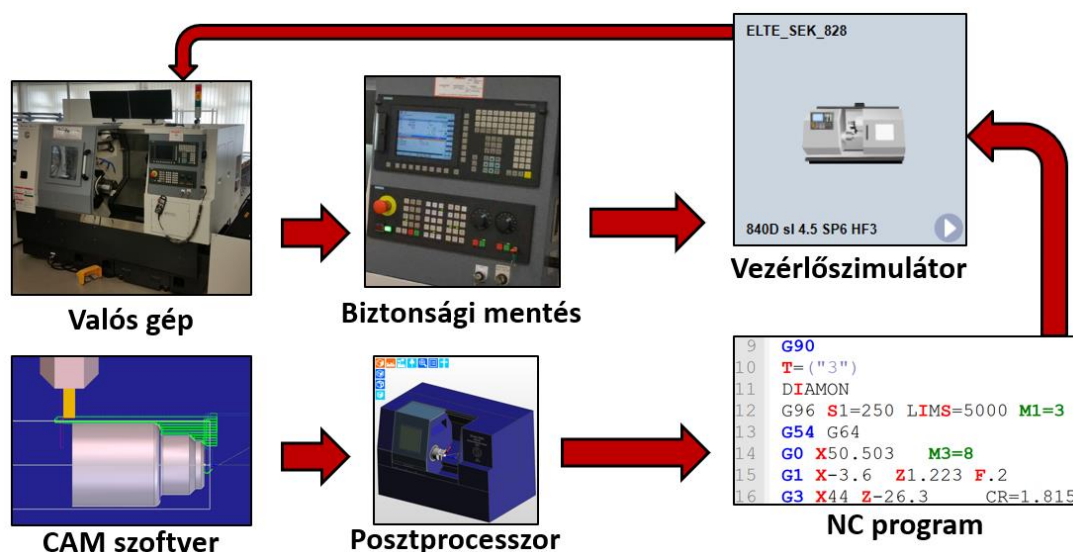
1. ábra: A szerszámgép és a vezérlője

a 2. ábra szemlélteti. A gyakorlatban ez azt jelenti, hogy a CNC gép vezérlőjéről készített biztonsági mentést importáltuk a Sinutrain vezérlőszimulátorba. Ezzel létrehoztuk a valós szerszámgép virtuális klónját, ami vezérlés szempontjából ugyanúgy viselkedik, ahogy azt a valós gép is tenné. Ebből következik, hogy minden programkód, ami a vezérlőszimulátorban helyesen futott, az a valós szerszámgépen is helyesen fut le. Azok a programkódok viszont, amelyek hibaüzenettel leálltak, a valós gépen is azonos hibaüzenettel álltak volna le.

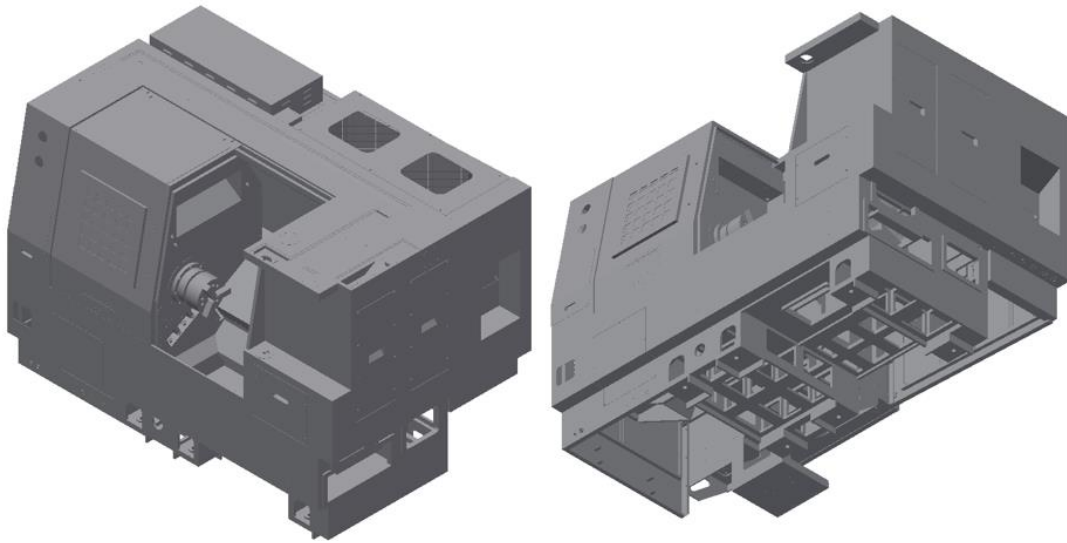
Ez azt jelenti a gyakorlatban, hogy a posztprocesszor által generált programkódok szintaktikai helyességéről pusztán a virtuális környezetben, a számítógépem segítségével (szerszámgép nélkül) meg tudunk bizonyosodni. További előny, hogy a szerszámgépet nem veszélyeztetjük az esetleges korai és hibás programokkal.

### 3. Fejlesztés főbb lépései

Posztprocesszor fejlesztésénél feltétlenül szükséges az adott szerszámgépről egy CAD modell. Ennek két fő kritériumnak kell megfelelnie: szimuláció biztonsága és futtathatóság. A CAM szoftverek egyik leghasznosabb funkciója a szimuláció – és annak a részeként az ütközésvizsgálat. Ennek segítségével a CAM szoftver figyelmeztetéssel megáll, ha ütközést észlel. Az ütközéseket a szerszámgép számos – előre definiált – alkatrészén végzi a szoftver. Annak érdekében, hogy ez helyesen működhessen, a szerszámgépről készült modell munkatérének teljes mértékben, minden irányban meg kell egyeznie a valós gépével. A másik fontos paraméter a modell fájlmérete, ugyanis a szimulációk és az ütközésvizsgálat nagy számításigényű folyamatok, amelyek a mai modern számítógépeket is képesek lehetnek túlterhelni, ezáltal nagymértékben lelassítani. Emiatt a fájlméretet célszerű megközelítőleg 5 megabyte környékére csökkenteni. A modell elkészítésére kétféle stratégia terjedt el. A konkrét gépről méréseket végezve felépíthető egy komplett új modell, azonban ezt ritkábban használják. Másik – amit jelen esetben alkalmaztunk, hogy a gép gyártójától kapott modellt kellő mértékben leegyszerűsítve létrehozható egy minden kritériumnak megfelelő új fájl. A 3. ábrán látható a gyártótól érkező modell. Ennek egyszerűsítése a számos bonyolult felületi elem miatt volt elkerülhetetlen. Az eredeti file 25,9 MB volt, ami jelentősen meghaladja a működési határt. Az egyszerűsítések során egyes alkatrészeket – amelyek az ütközésszimuláció szempontjából nem relevánsak – egyszerűen kitöröltük, míg az alkatrészek többségének geometriáját egyszerűsítettük. Ennek során eltüntettük a felesleges furatokat, a csapágyakat, kötőelemeket és egyéb gépelemeket, felületeket.



2. ábra: Kiber-fizikai fejlesztés folyamatábrája



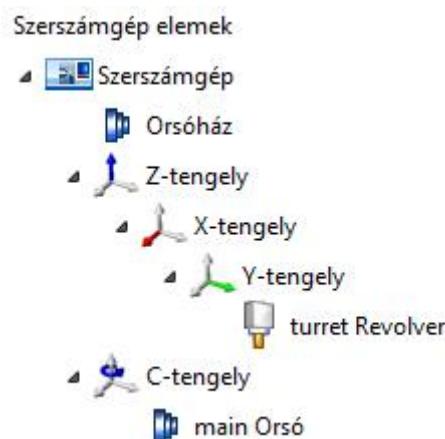
3. ábra: A gyártótól érkező CAD modell

A posztprocesszor az Edgecam Kódvarázsló nevű menüpontjában készíthető el. A posztprocesszorok többsége sablon alapján készül. A létrehozásnál fel is kínál a szoftver számos vezérlőtípust. Ebben az esetben azonban a Sinumerik 828D nem szerepelt a listában, így a legjobban hasonlító, 840-es sablont választottuk. Ezután lehetséges az elkészített modell alapján az Edgecam-ben kinematikai szempontból is felépíteni a szerszámgépet, a 4. ábrán látható hierarchiának megfelelően. A modell importálásával és a valós kinematikai adatok bevitelével elkészült az ütközésvizsgálat szimulációs környezete.

A posztprocesszor elkészítésének utolsó lépése, hogy az adott gépnek megfelelően kitöltsük a paramétereket, melyek a G-kód szintaktikáját fogják meghatározni. Ez a folyamat a legidőigényesebb és ezért a legköltségesebb. További probléma, hogy hagyományos módszerrel a CNC gép ezen idő alatt nem termel – így a kieső termelés plusz költségként jelentkezik.

#### 4. Tesztelés kiber-fizikai rendszerben

A gépen futó és a sablonból választható két vezérlő verzió közt a gyakorlatban elenyésző különbségek vannak, azonban még helyes sablonválasztás esetén is gyakorlatilag kizárt, hogy egy posztprocesszor



4. ábra: Posztprocesszor modellfa



elsőre helyesen kezdjen működni. A hibakeresésre létrehoztuk egy alkatrész 3D modelljét. Erre kezdünk el megmunkálási programokat készíteni az Edgcam-ben, majd a megmunkálási programokat gépi programkódra fordítottattuk a posztprocesszorral. Az így generált kódot a vezérlőszimulátorral teszteltük. Megszokott dolog, hogy az első próbálkozások rendre kudarcba fulladtak – a programok hibaüzenettel leálltak (5. ábra). Azonban ezek segítségével lehetett következtetni a hibák forrására. Mivel a virtuális gép teljesen azonosan működik a valós géppel, az itt kapott hibaüzenet – és hibakód – alapján a Siemens programozói kézikönyvben kikereshető, hogy mi okozta a hibás futást. A programozói kézikönyvből kikereshető, hogy mi miatt volt rossz a generált kód. Azonban az adott generált kódrészlet megváltoztatásához, az esetek többségében nem egyértelmű, hogy mit kell megváltoztatni a posztprocesszorban. A CAM rendszerek tetszőlegesen bonyolult geometriákra képesek megmunkálási programokat írni, azonban az ezekből generált kódok a valóságban sokszor tagolatlanok, nehezen átláthatók és több száz sorosak lehetnek egy egyszerű alkatrész esetén is. A posztprocesszorban az egyes műveletek generálásért külön alrendszerek felelősek (pl. lineáris interpoláció, szerszámcsere vagy forgácsolás fűróciklus kódszerkesztője). Egy egyszerű esztergálás végrehajtásánál is szükség van szerszámcsere, lineáris interpolációra – előtolással és gyorsjáráttal egyaránt, hazaállásra stb. Ezáltal a generált programkódot nézve nem egyértelmű, hogy egy adott programrészlet generálásáért konkrétan melyik kódszerkesztő a felelős. Ennek felderítésére használható az Edgcam egyik diagnosztikai funkciója – a Teljes kódkészítő követés generálása. Ennek hatására a generált kódban megjelennek kiemelt sorok (6. ábra), amelyek leírják, hogy az adott kódrészlet megváltoztatására melyik kódszerkesztőt kell módosítani. Miután a posztprocesszor olyan kódokat generált, amelyek a vezérlőszimulátorban hibátlanul futottak, következhetek a valós gépen történő tesztek.

## 5. Valós tesztek és eredmények

A valós szerszámgépen történő tesztelés rávilágított, hogy a valós tesztelést sosem szabad elhanyagolni, ugyanis a virtuális környezetben jól működő posztprocesszor a valós gépen lezajlott első teszt során hibás munkadarabot (7. ábra) és törött lapkát eredményezett. Ennek oka az volt, hogy a posztprocesszorban a számos paraméter közül a főorsó forgásiránynál 3 helyett 4 volt megadva, ezáltal  $M3$  helyett  $M4$  kóddal generálta a főorsó forgását, így az fordítva forgott. Az érték korrigálása után a program helyesen futott és a gép sikeresen megmunkálta a tervezett alkatrészt. A selejtes és a pontos munkadarab a 7. ábrán látható.

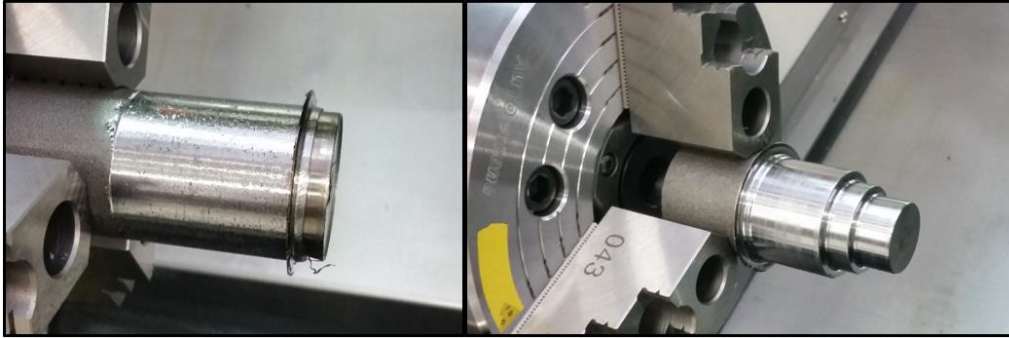
A fejlesztési módszernek köszönhetően megtakarított üzemórákat konkrétan számszerűsíteni nem könnyű feladat. Kezdetben számos hibaüzenet, sikertelen futtatás előzte meg a működő programokat, valamint egyes funkciók működőképessé tétele több napot vett igénybe. A funkciók többségéhez 8-10 mintaprogram is készült, mire végül jól működtek. A több száz tesztprogram lefuttatása, majd az ezek szerint történő korrigálás összesített munkaóráinak száma nagyságrendileg eléri a 100-at.



5. ábra: Hibaüzenet és a hozzá tartozó hibakód



6. ábra: Hibakeresés a generált programkódban

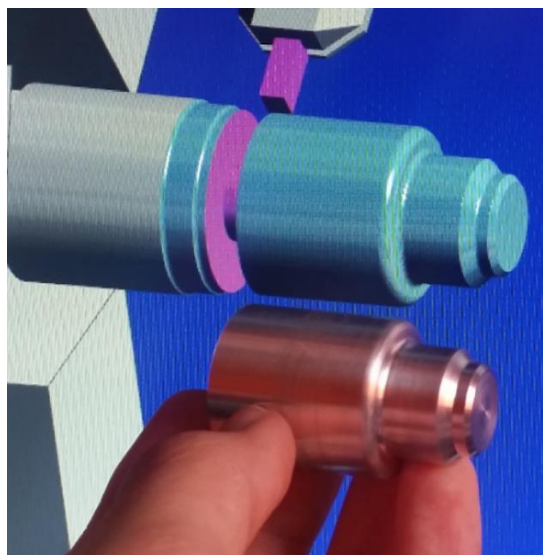


**7. ábra:** Első valós teszt eredményei

Gyakorlatilag ennyi tesztelési üzemórától biztosan mentesült a valós szerszámgép, ami jelentős megtakarítást jelent, hiszen a CNC gép ez alatt az idő alatt is termelt. Ezen felül még fontosabb előny, hogy a gép a tesztelés során kevésbé volt terhelve és veszélyeztetve a hibás programok miatt. Ezek a mai, egyre jobban kiélezett gyártói versenyben egy termelő vállalat számára nem elhanyagolható szempontok.

## 6. Összefoglalás

A munkánk során elkészítettük az esztergagép leegyszerűsített modelljét és létrehoztunk egy 2-tengelyes esztergálási funkciókkal működő Edgcam-es posztprocesszort, amelynek funkcionalitását teszt munkadarabok elkészítésével is igazoltuk. A tesztelési módszernek köszönhetően számos gépi üzemórát tudtunk megtakarítani, valamint megkíméltük a szerszámgépet az ütközések esélyétől is. A CPS rendszerben történő fejlesztés tapasztalatai, hogy a vezérlőszimulátorban történő tesztelés kiválóan alkalmas a programozási szintaktikai hibák korai stádiumú kiszűrésére, azonban egyes jelenségek a szimulátorban nehezen – vagy akár egyáltalán nem – érzékelhetők helyesen. Ezek főként a valós mozgásokhoz kapcsolódnak, például a forgási irányok és a mozgási sebességek. Az elért eredményt a 8. ábra mutatja be.



**8. ábra:** Teszt alkatrész az Edgcam-ben és a valóságban

## 7. Köszönetnyilvánítás

Az ED\_18-1-2019-0030 szerződésszámú projekt (Alkalmazásiterület-specifikus nagy megbízhatóságú informatikai megoldások tématerület) a Nemzeti Kutatási Fejlesztési és Innovációs Alapból biztosított támogatással, a Tématerületi kiválósági program támogatásával valósult meg.

## 8. Irodalomjegyzék

- [1] Mátyási G. és Sági G., *Számítógéppel támogatott technológiák*, Műszaki Könyvkiadó Kft., Budapest, 2009.
- [2] Boór F., Hervay P., Kakuk Gy., Markos S., Mátyási G. és Mikó B., *CAM Tankönyv*, Typotex Kiadó, Budapest, 2012.